CSL7640: Natural Language Understanding

Lecture Scribing Notes

Instructor: **Anand Mishra**
IIT Jodhpur

Scribe: Suraj Kumar (M25CSA030)

Lecture #: 7 — Date: 3-2-26

Lecture Overview

- **Topic(s):** Hidden Markov Models (HMM), Viterbi Algorithm
- **Pre-requisites:** Probability theory, conditional probability, Markov chains
- **Reading Material:** Jurafsky & Martin, Chapter on HMMs

Learning Objectives

- Understand the structure and assumptions of Hidden Markov Models
- Learn how sequential data is modeled probabilistically
- Derive and apply the Viterbi algorithm for decoding
- Analyze the computational efficiency of Viterbi decoding

1 Introduction

Many natural language processing tasks involve sequential data, such as sentences, speech signals, or biological sequences. Hidden Markov Models (HMMs) provide a probabilistic framework to model such sequences when the underlying states are not directly observable.

In HMMs:

- The system evolves through a sequence of hidden states
- Each hidden state probabilistically generates an observable output

The Viterbi algorithm is used to infer the most likely hidden state sequence for a given observation sequence.

2 Key Concepts

2.1 Hidden Markov Model

An HMM is defined by the parameter set:

$$\lambda = (S, O, A, B, \pi)$$

where:

- $S = \{s_1, s_2, \ldots, s_N\}$ is the set of hidden states
- $O = \{o_1, o_2, \ldots, o_M\}$ is the observation vocabulary
- $A = [a_{ij}]$ is the transition probability matrix
- $B = [b_j(o)]$ is the emission probability distribution
- $\pi = [\pi_i]$ is the initial state distribution

2.2 Markov Assumptions

First-order Markov property:

$$P(q_t \mid q_{t-1}, q_{t-2}, \ldots, q_1) = P(q_t \mid q_{t-1})$$

Output independence assumption:

$$P(o_t \mid q_t, q_{t-1}, \ldots) = P(o_t \mid q_t)$$

These assumptions make inference computationally tractable.

3 Mathematical Formulation

The joint probability of a hidden state sequence $q_1, \ldots, q_T$ and observation sequence $o_1, \ldots, o_T$ is:

$$P(q_{1:T}, o_{1:T}) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}q_t} \prod_{t=1}^{T} b_{q_t}(o_t)$$

The decoding problem is defined as:

$$q^*_{1:T} = \arg\max_{q_{1:T}} P(q_{1:T} \mid o_{1:T})$$

Since $P(o_{1:T})$ is constant, this reduces to maximizing the joint probability.

4 The Viterbi Algorithm

The Viterbi algorithm solves the decoding problem using dynamic programming.

4.1 Viterbi Variables

$$\delta_t(j) = \max_{q_{1:t-1}} P(q_{1:t-1}, q_t = s_j, o_{1:t})$$

$$\psi_t(j) = \arg\max_i \left[\delta_{t-1}(i) a_{ij}\right]$$

4.2 Initialization

$$\delta_1(j) = \pi_j b_j(o_1), \quad \psi_1(j) = 0$$

4.3 Recursion

For $t = 2, \ldots, T$:

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t)$$

4.4 Termination and Backtracking

$$q_T^* = \arg\max_j \delta_T(j)$$

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad \text{for } t = T-1, \ldots, 1$$

5 Examples

- Part-of-speech tagging of the sentence: *"The fans watch the race"*
- States correspond to POS tags, observations are words
- Viterbi efficiently computes the best tag sequence

6 Figures / Diagrams

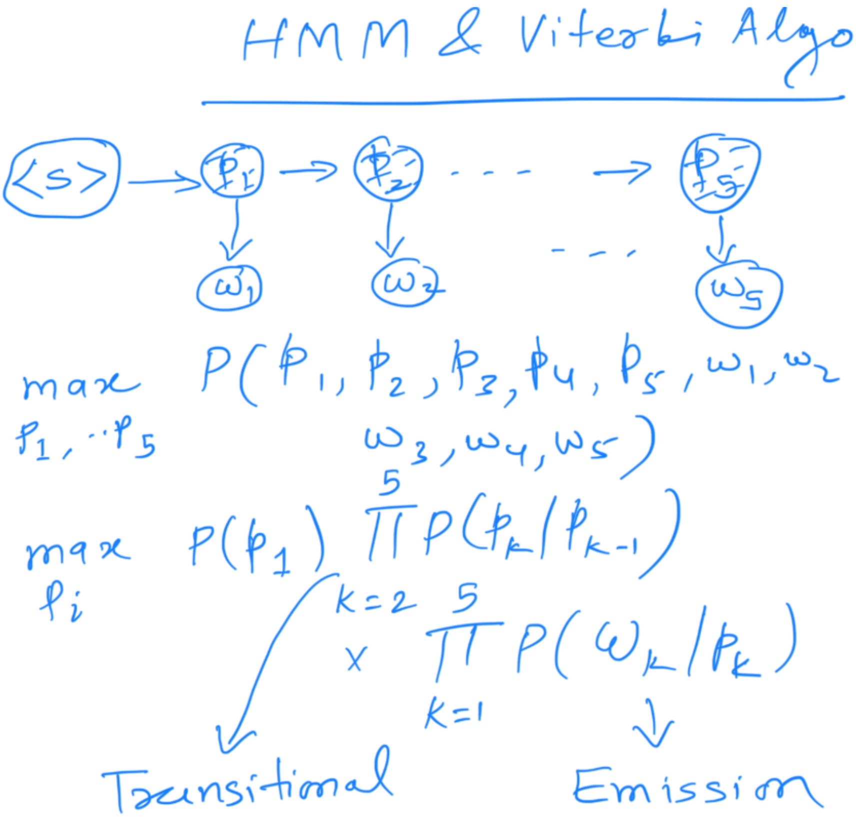
HMM & Viterbi Algo
<s>
p_1
p_2
p_5
w_1
w_2
w_5
$\max_{p_1, \ldots p_5} P(p_1, p_2, p_3, p_4, p_5, w_1, w_2, w_3, w_4, w_5)$
$\max_{p_i} P(p_1) \prod_{k=2}^{5} P(p_k / p_{k-1})$
$\times \prod_{k=1}^{5} P(w_k / p_k)$
Transitional
Emission

Figure 1: HMM and Viterbi Algo

7 Class Discussion / Insights

- Why greedy decoding fails for sequence labeling
- Importance of dynamic programming in structured prediction
- Trade-off between model simplicity and expressive power

8 Summary

- HMMs model sequential data with hidden states
- Viterbi algorithm finds the most likely hidden state sequence
- Complexity is polynomial rather than exponential

9 Open Questions / To Think About

- How do HMM assumptions limit modeling power?
- How do CRFs and neural models overcome these limitations?

References

- Jurafsky, D., & Martin, J. *Speech and Language Processing*
- Rabiner, L. (1989). A Tutorial on Hidden Markov Models